

Отчёт по лабораторной работе №1

дисциплина: Информационная безопасность

Зорин Илья Михайлович

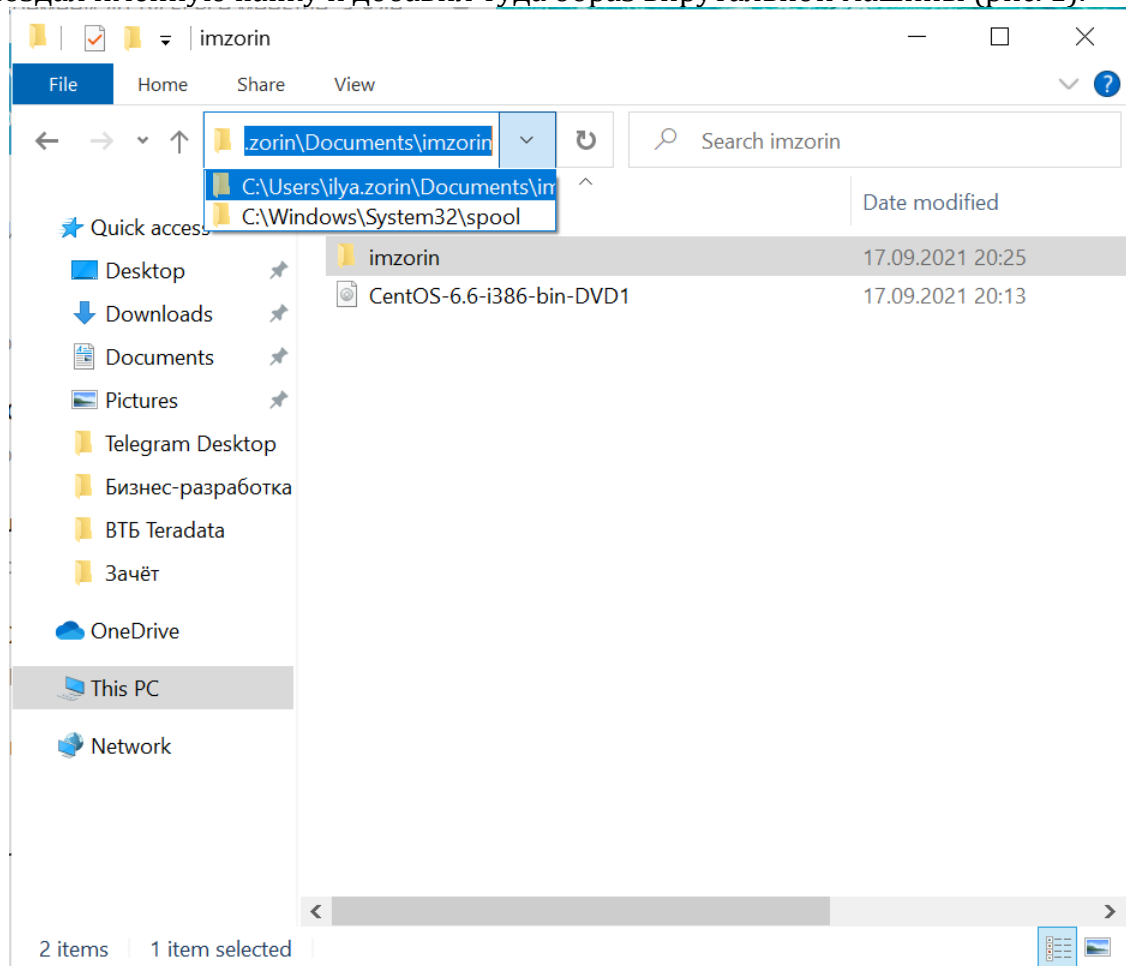
Содержание

Цель работы

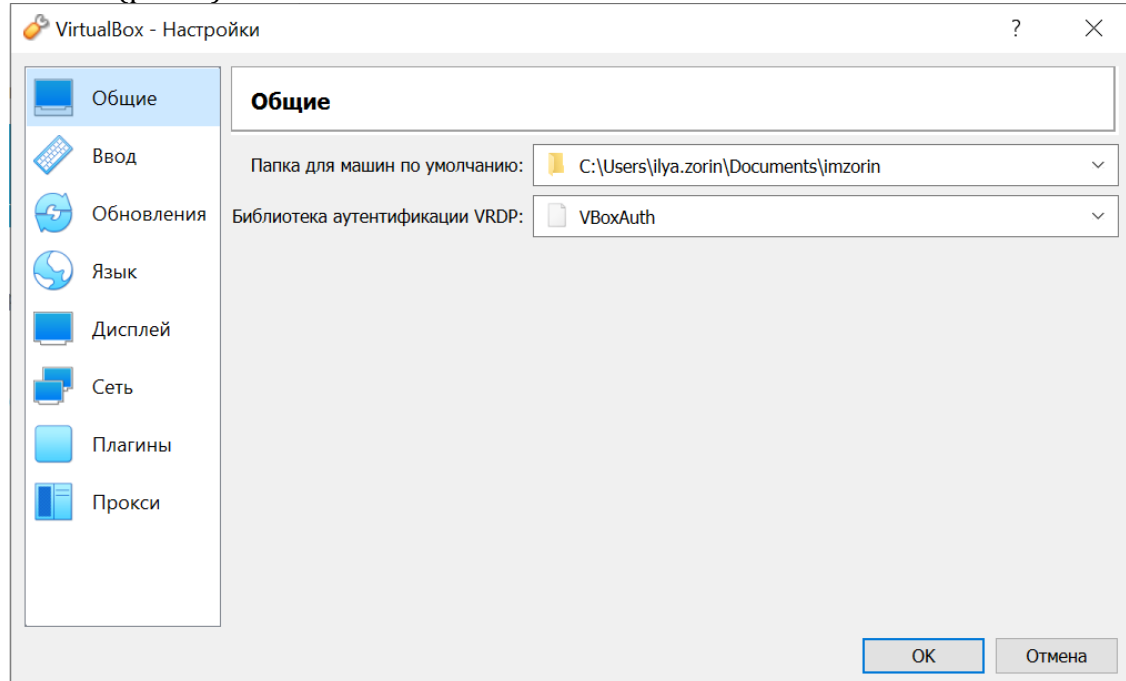
- Приобретение практических навыков установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.
- Изучить идеологию и применение средств контроля версий.
- Научиться оформлять отчёты с помощью легковесного языка разметки Markdown.

Выполнение лабораторной работы

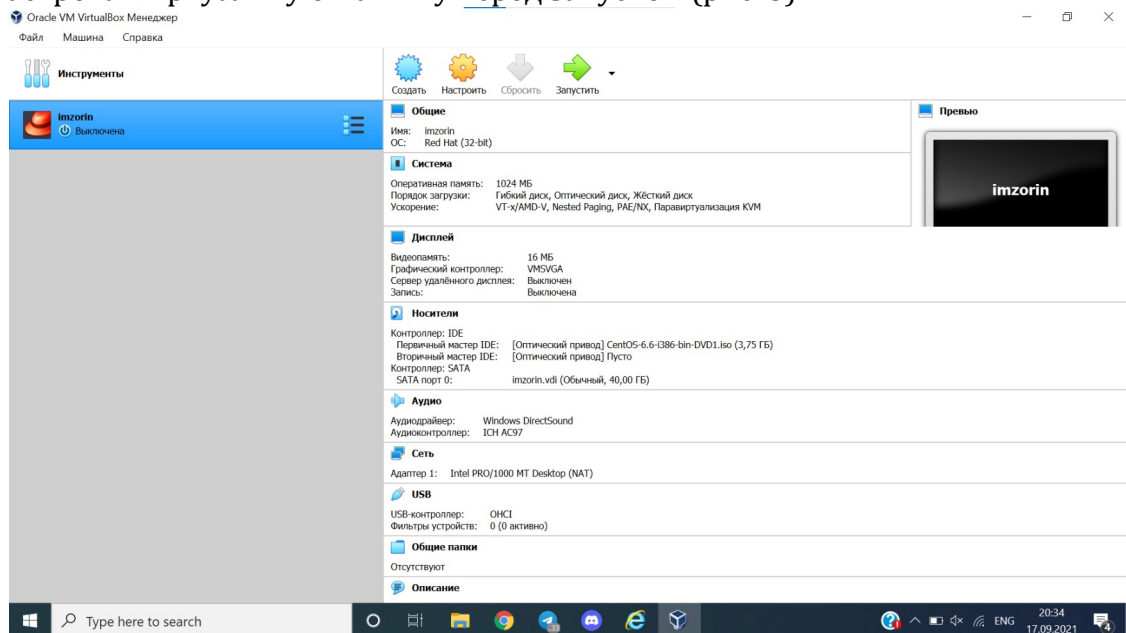
1. Создал именную папку и добавил туда образ виртуальной машины (рис. 1).



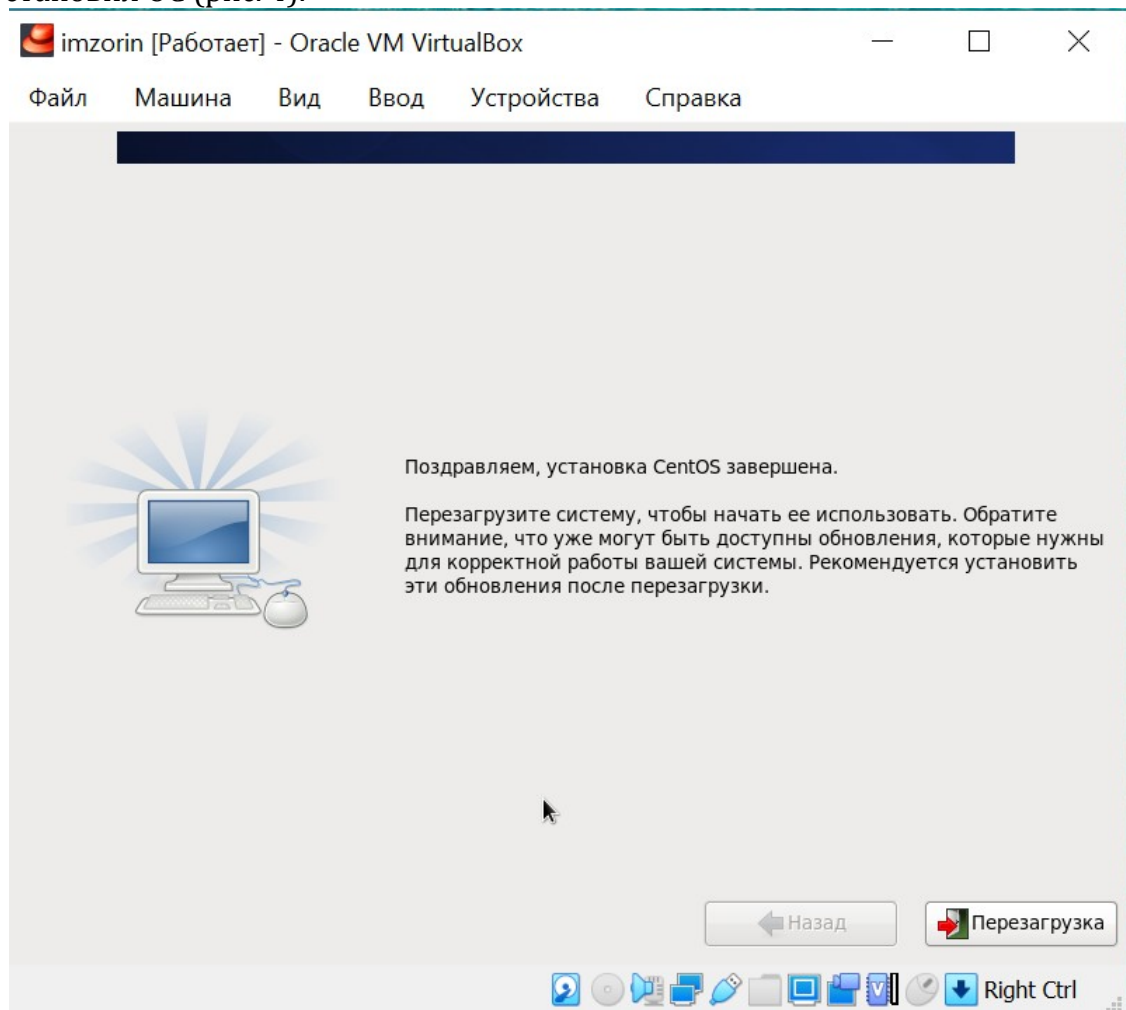
2. Изменил папку по умолчанию для расположения каталога виртуальной машины (рис. 2).



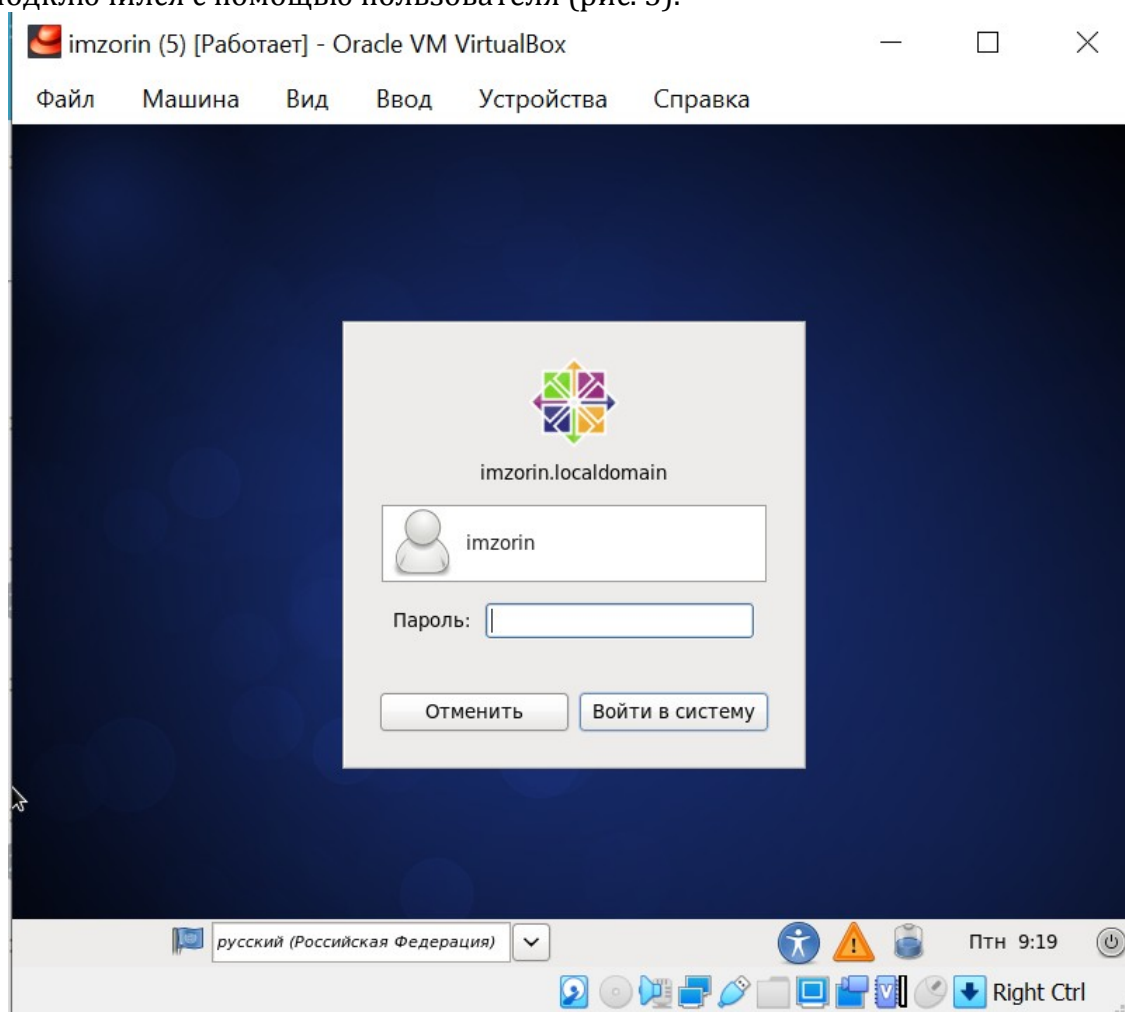
3. Настроил виртуальную машину перед запуском (рис. 3).



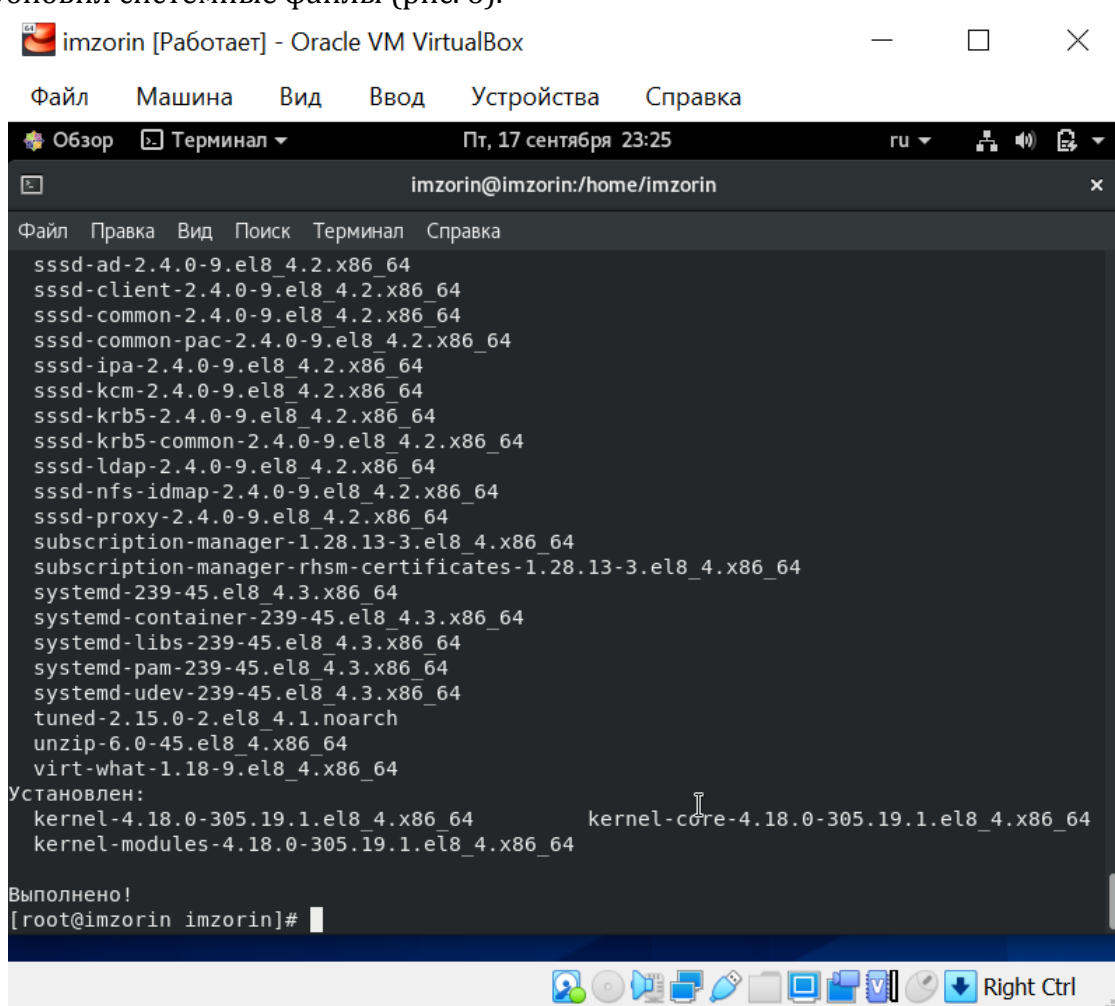
4. Установил ОС (рис. 4).



5. Подключился с помощью пользователя (рис. 5).



6. Обновил системные файлы (рис. 6).



```
imzorin [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка

Обзор  Терминал  Пт, 17 сентября 23:25  ru  [иконки]

imzorin@imzorin:/home/imzorin
Файл  Правка  Вид  Поиск  Терминал  Справка

sssd-ad-2.4.0-9.el8_4.2.x86_64
sssd-client-2.4.0-9.el8_4.2.x86_64
sssd-common-2.4.0-9.el8_4.2.x86_64
sssd-common-pac-2.4.0-9.el8_4.2.x86_64
sssd-ipa-2.4.0-9.el8_4.2.x86_64
sssd-kcm-2.4.0-9.el8_4.2.x86_64
sssd-krb5-2.4.0-9.el8_4.2.x86_64
sssd-krb5-common-2.4.0-9.el8_4.2.x86_64
sssd-ldap-2.4.0-9.el8_4.2.x86_64
sssd-nfs-idmap-2.4.0-9.el8_4.2.x86_64
sssd-proxy-2.4.0-9.el8_4.2.x86_64
subscription-manager-1.28.13-3.el8_4.x86_64
subscription-manager-rhsm-certificates-1.28.13-3.el8_4.x86_64
systemd-239-45.el8_4.3.x86_64
systemd-container-239-45.el8_4.3.x86_64
systemd-libs-239-45.el8_4.3.x86_64
systemd-pam-239-45.el8_4.3.x86_64
systemd-udev-239-45.el8_4.3.x86_64
tuned-2.15.0-2.el8_4.1.noarch
unzip-6.0-45.el8_4.x86_64
virt-what-1.18-9.el8_4.x86_64
Установлен:
kernel-4.18.0-305.19.1.el8_4.x86_64      kernel-core-4.18.0-305.19.1.el8_4.x86_64
kernel-modules-4.18.0-305.19.1.el8_4.x86_64

Выполнено!
[root@imzorin imzorin]#
```

Выводы

- Приобрёл практические навыки установки операционной системы на виртуальную машину, настройки минимально необходимых для дальнейшей работы сервисов.
- Изучил идеологию и применение средств контроля версий.
- Научился оформлять отчёты с помощью легковесного языка разметки Markdown.

Ответы на контрольные вопросы

1. Что такое системы контроля версий (VCS) и для решения каких задач они предназначены?
 - Это - программное обеспечение для облегчения работы с изменяющейся информацией.
2. Объясните следующие понятия VCS и их отношения:

- хранилище - место, где система управления версиями хранит все документы вместе с историей их изменения и другой служебной информацией,
 - *commit* - дерево и некая дополнительная информация,
 - история - данные о версиях файла,
 - рабочая копия - актуальная версия файла.
3. Что представляют собой и чем отличаются централизованные и децентрализованные VCS? Приведите примеры VCS каждого вида.
 - Централизованные VCS имеют модель клиент-сервер: один центральный репозиторий, с которым разработчики взаимодействуют по сети. В отличие от централизованной модели, децентрализованная модель может существовать несколько экземпляров репозитория, которые время от времени синхронизируются между собой.
 - Пример централизованной VCS - SVN. Одна из самых распространенных систем контроля версий. Для децентрализованной модели примером является *git*.
 4. Опишите действия с VCS при единоличной работе с хранилищем.
 - **
 5. Опишите порядок работы с общим хранилищем VCS.
 - **
 6. Каковы основные задачи, решаемые инструментальным средством *git*?
 - Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом.
 - Также они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви.
 7. Назовите и дайте краткую характеристику командам *git*.
 - С основными командами можно ознакомиться [тут](#)
 8. Приведите примеры использования при работе с локальным и удалённым репозиториями.
 - При создании локального репозитория достаточно сначала сделаем предварительную конфигурацию, указав имя и email владельца репозитория:
 - *git config --global user.name "Имя Фамилия"*
 - *git config --global user.email "work@mail"*
 - При работе с удалёнными репозиториями необходимо сначала зарегистрироваться на сервере, например, *github*.
 9. Что такое и зачем могут быть нужны ветви (branches)?
 - Ветвь — направление разработки, независимое от других. Ветвь представляет собой копию части (как правило, одного каталога) хранилища, в которую можно вносить свои изменения, не влияющие на другие ветви. Документы в разных ветвях имеют одинаковую историю до точки ветвления и разные — после неё.
 10. Как и зачем можно игнорировать некоторые файлы при *commit*?

- *Во время работы над проектом так или иначе могут создаваться файлы, которые не требуется добавлять в последствии в репозиторий. Например, временные файлы, создаваемые редакторами, или объектные файлы, создаваемые компиляторами. Можно прописать шаблоны игнорируемых при добавлении в репозиторий типов файлов в файл .gitignore с помощью сервисов.*